Guiding-DIP-Early-Stopping-with-DDPM-inspired-Supervision

Introduction

本報告涵蓋基於 DIP 基礎上增加 DDPM 噪音方法的實作，藉以引導 DIP 學習噪音的特徵。第一部分 Implementation Process (I)會分析原始 DIP 與加上噪音所產生的問題，第二部分 Implementation Process (II)則解決問題與最終結果。第三部分 Further Analysis 則是分析 DIP with Guiding Noisy Image 在不同程度噪音圖片的表現。

Experiment Setup

Dataset:Mammals Image Classification Dataset (45 Animals) on Kaggle

Model

Architecture: 類似 Unet 但不是標準 Unet 的 Hourglass

Framework: PyTorch

Optimizer: Adam

Loss: Mean Squared Error (MSE)

Training Procedure

DIP only

DIP with Guiding Noisy Image

Adding Noise Method

使用的是 DDPM 的前向過程。在前向過程中，從正態高斯分佈中取樣的 Noise 會逐漸添加到圖像樣本中。給定圖像 x_0 和 Noise ϵ，其中 $\epsilon \sim N(\mu, I)$，在包含 T 個時間步的 Markov chain，從 t=0 到 t=T-1，前向過程 q 可以被定義為：

$$q(x_{t+1}|x_t) = N(x_{t+1}; \sqrt{1-\beta_{t+1}}x_t, \beta_{t+1}I)$$

其中 β 由原始 DDPM 論文的 beta-min 和 beta-max 定義的值，定義方式如下：

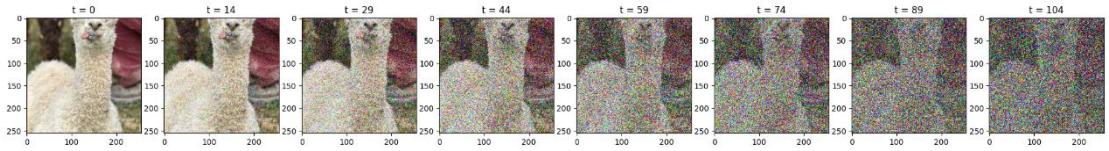
線性方法：$\beta_t = \beta_{\min} + t \cdot \frac{\beta_{\max} - \beta_{\min}}{T}$

餘弦方法：$\beta_t = \beta_{\min} + 0.5(\beta_{\max} - \beta_{\min})(1 - \cos(\frac{t}{T}\pi))$

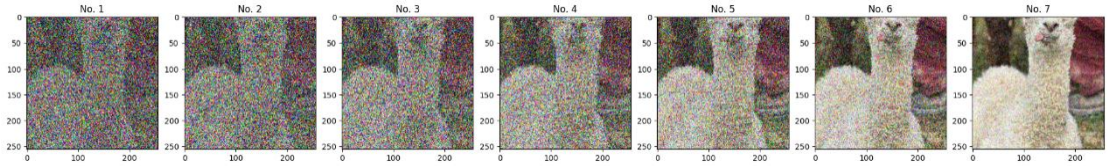
本實驗過程經過測試，最終採用的方法為線性。

Implementation Process (I)

先透過 DDPM 前向過程，產生數張 noisy 圖片 ，t=0 為無 noisy。

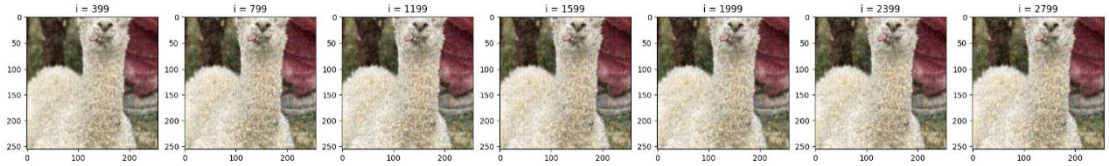


根據作業要求，由 noise 最嚴重的圖片開始，依序放入 DIP 過程。因此將 nosiy 圖片序列翻轉，以及移除無 noisy 圖片 。總共 7 張圖片

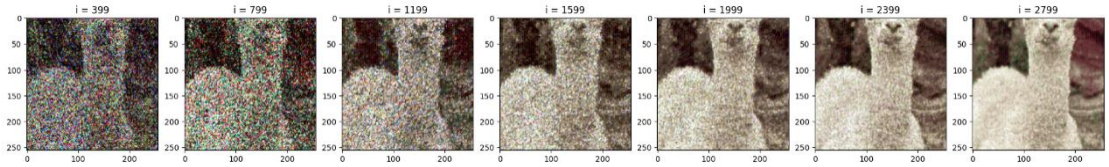


每張圖片跑 400 次迭代，總共 2800 次。

原始 DIP 在整個過程只跑第 7 張（Noise 最少的圖片），結果如下。



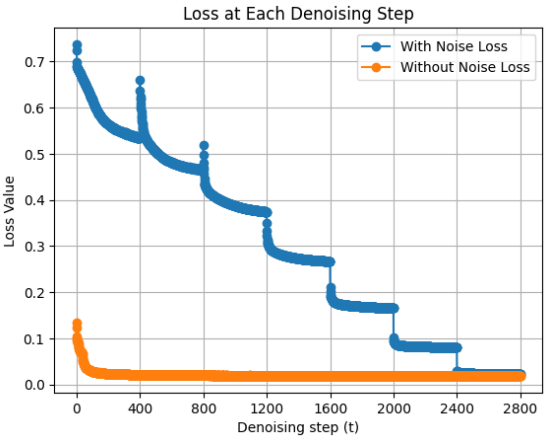
DIP with Guiding Noisy Image 在不同 400 次迭代的過程跑不同 noisy 程度的圖片 ，由最嚴重到最輕，結果如下。



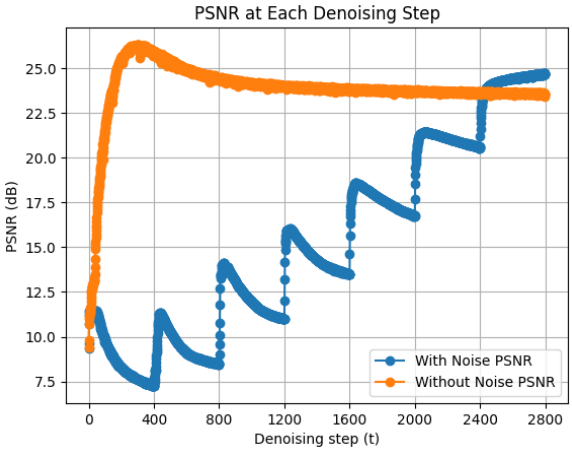
Analysis of Implementation Process (I)

原始 DIP 為橘色線，DIP with Guiding Noisy Image 為藍色線。

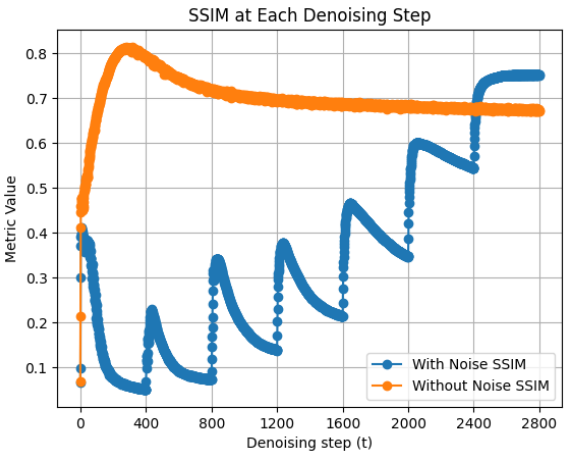
右圖為兩者 Loss 的比較圖，計算 Loss 方式是產生的圖片比對原圖（無 noise）。可看出有 Noise 的過程在每次換圖片時，loss 會回到高點。



右圖為兩者的 PSNR 比較圖。PSNR 是用來評估圖像訊號的差異，因此可用來檢視噪音是否被正確消除。以每 400 迭代為一個區間，可發現當 DIP with Guiding Noisy Image 經過一定的 t，會有個早停點，之後 PSNR 開始下降，出現過擬和（Overfitting），表示噪音特徵被過度重視。也可發現原始 DIP 需要大約 300 次的疊代才能抵達高峰。



右圖為兩者的 SSIM 比較圖。SSIM 是用來評估人眼對圖像感知的差異，包含亮度、對比度和結構。以每 400 迭代為一個區間，可發現當 DIP with Guiding Noisy Image 經過一定的 t，SSIM 開始下降，一樣也是出現過擬和（Overfitting）。



根據分析內容，為了解決過擬和與效果不如原始 DIP。於是接續 Implementation Process (II)。

Implementation Process (II)

這裡提出三個調整方法有效改善 DIP with Guiding Noise 的表現。

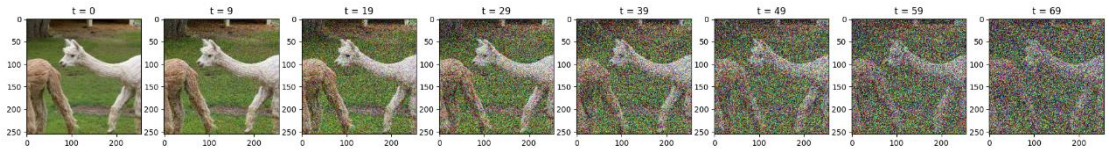
Adjusting Learning rate

原先原始 DIP 與 DIP with Guiding Noise 的 learning rate 都是 0.01，但經過多次測試後，認為這導致過度關注噪音，導致表現變差，因此將 DIP with Guiding Noise 的 learning rate 調降至 0.0085，原始 DIP 仍然是 0.01。

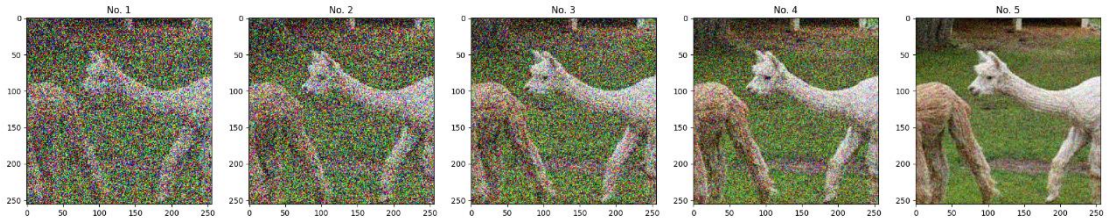
Reduce the number of given noisy image

原本噪音最嚴重的圖片幾乎看不太到輪廓，於是認為噪音過多，無法正確學習噪音特徵。因此除了降低每次噪音的增加量，並且捨去噪音太嚴重的圖片。

以下是前向過程產生的噪音圖片序列：



經過刪減後的圖片序列（Implementation Process (I)使用 7 張，這裡為 5 張）：



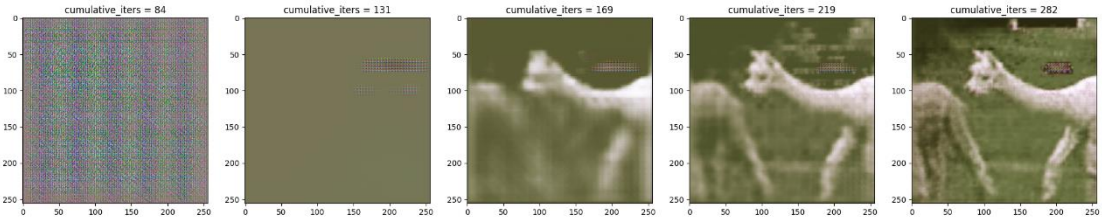
Reduce the number of iterations based on PSNR

原本每張圖片跑 400 次迭代，沒有善用早停點，導致 overfitting。因此增加停止點，當 PSNR 下降超過 10 次，立即跳至下一階段圖片的迭代，因此不同階段的迭代次數會不同。公平起見，原始 DIP 也會跑相同的迭代次數。

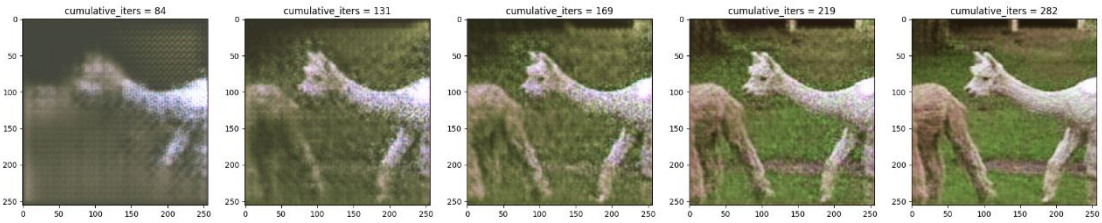
Result

總共實驗總共 5 個階段，迭代次數分別為 84,47,38,50,63，總次數為 283 次。

原始 DIP 只跑第 5 張圖片（噪音最少）結果如下：



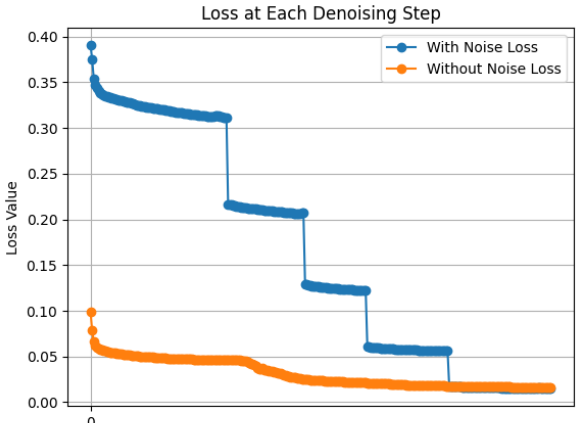
DIP with Guiding Noisy Image 在不同 400 次迭代的過程跑不同 noisy 程度的圖片，由最嚴重到最輕，結果如下。

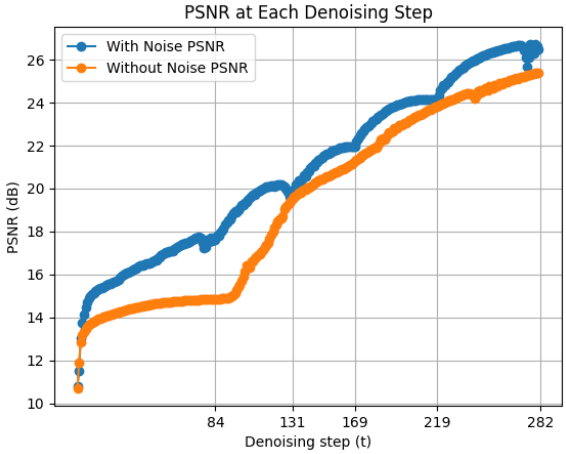


Analysis of Implementation Process (II)

原始 DIP 為橘色線，DIP with Guiding Noisy Image 為藍色線。

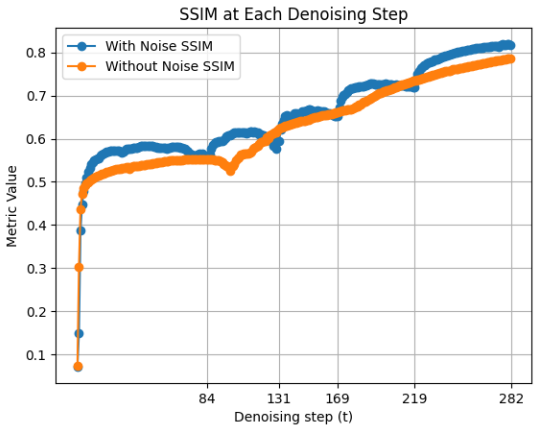
與 Implementation Process (I)的結果大致相同，因此不多說明。





可以看出 DIP with Guiding Noise Image 在每個階段經過高峰後的下降曲線大幅縮短，以及**總迭代次數約為 Implementation(I)的 1/10 倍**，這是因為紀錄 PNSR 並設立早停點的結果。也可看出**在原始 DIP 抵達高峰之前，DIP with Guiding Noise Image 就先抵達高峰。**

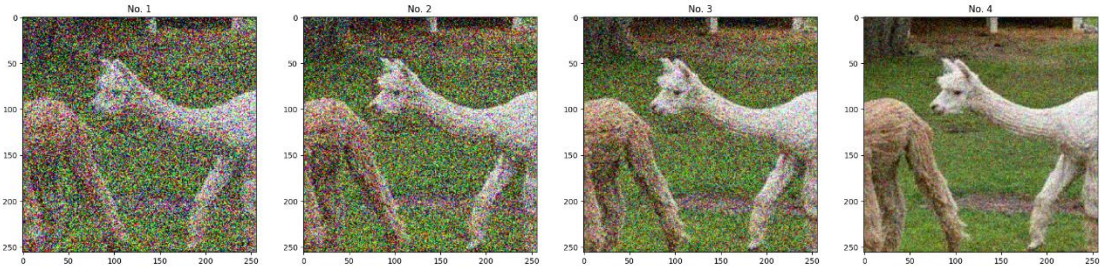
右圖為兩者的 SSIM 比較圖，可看出兩者到最後仍然沒到達最高峰。這也是根據 PSNR 設立早停點的小缺點。



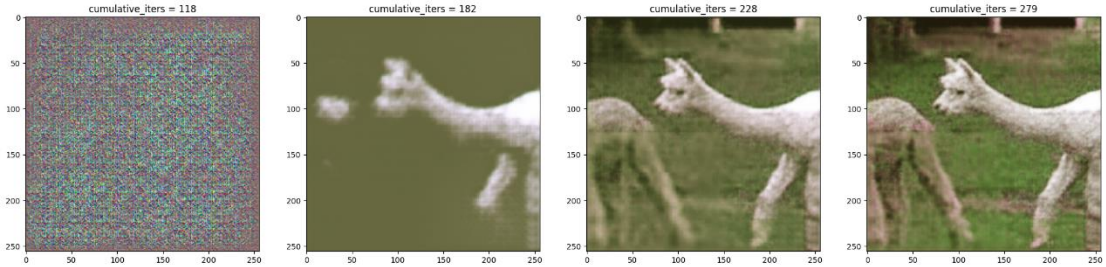
Further Analysis

為了分析 DIP with Guiding Noisy Image 在不同噪音程度的表現，因此在給定的噪音 image 中，分成四張噪音多的圖片與四張噪音少的圖片 。

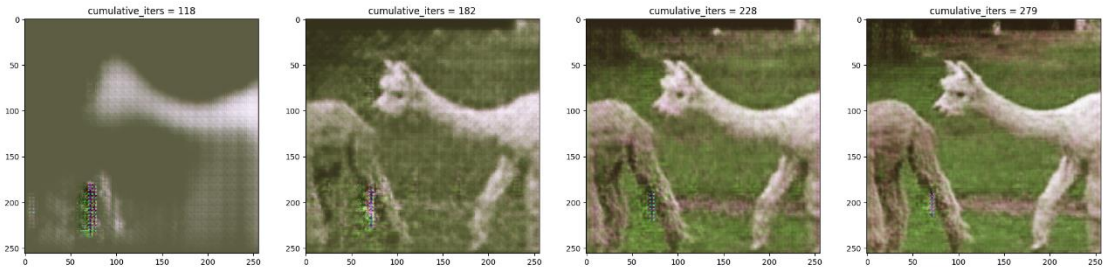
在四張噪音少圖片的 case：



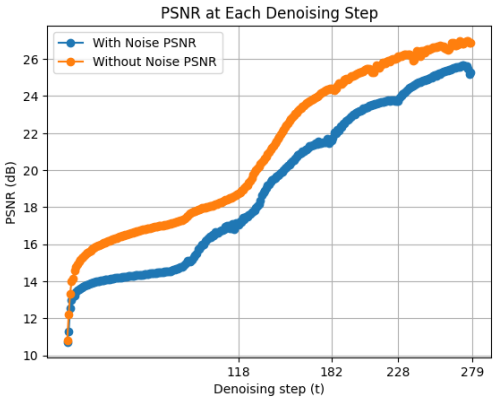
原始 DIP 結果：

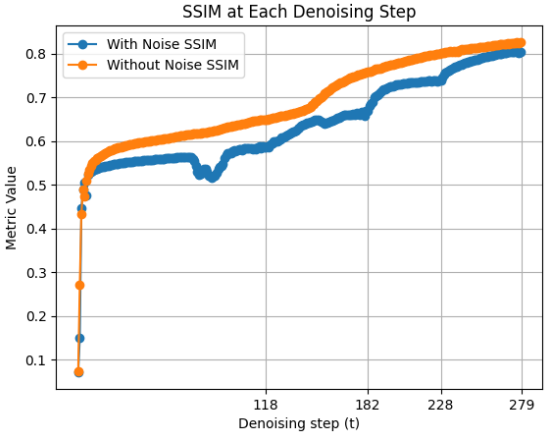


DIP with Guiding Noisy Image 的結果：

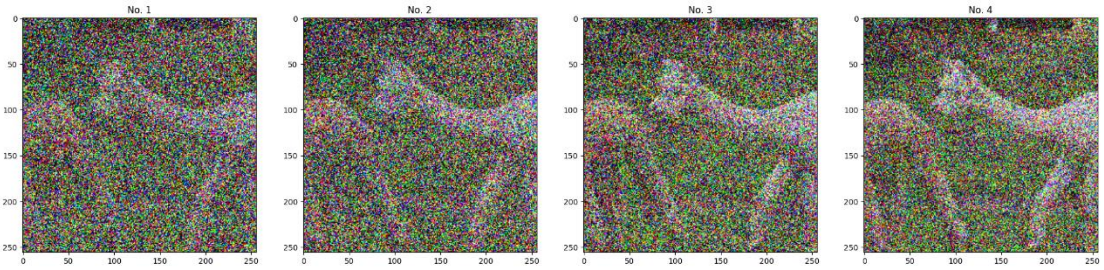


PSNR 與 SSIM

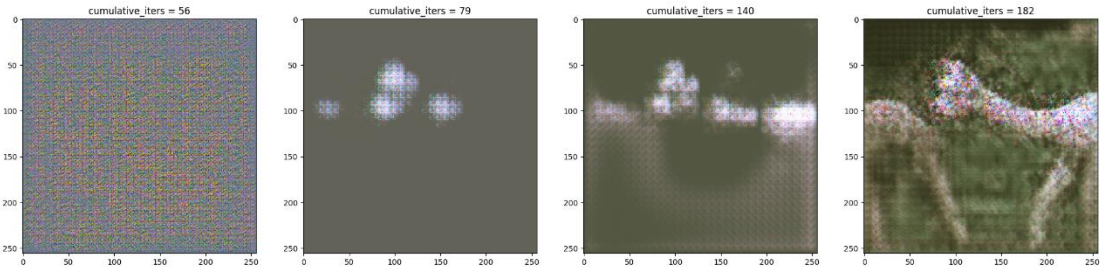




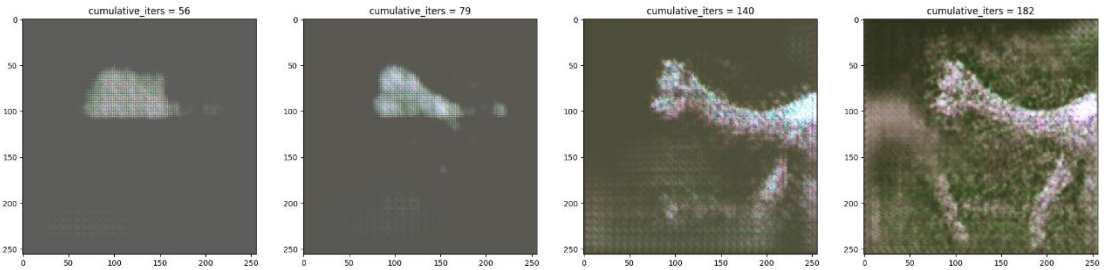
在四張噪音多圖片的 case：



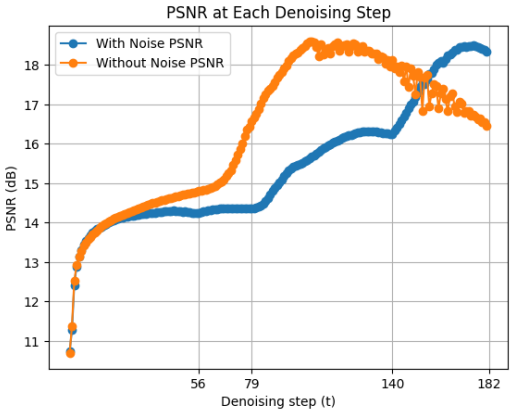
原始 DIP 的結果：

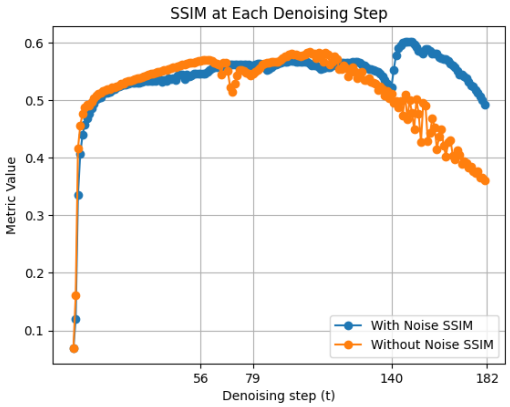


DIP with Guiding Noisy Image 的結果：



SNR 與 SSIM





Conclusion：

透過 Implementation Process (I)和 Implementation Process (II)可以得知使用 Guiding Noisy Image 的方法引導 DIP 加快且正確去除噪音是可行的。但是透過 Further Analysis 也能得知，Guiding Noisy Image 在噪音多的圖片比較能夠發揮，而在正常噪音少的圖片，表現仍差於原本的 DIP。

Reference：

Mammals Image Classification Dataset (45 Animals)-
https://www.kaggle.com/datasets/asaniczka/mammals-image-classification-dataset-45-animals?select=mammals

DeepImagePrior-

https://github.com/safwankdb/Deep-Image-Prior/tree/master

Denoising Diffusion Probabilistic Model-

https://www.kaggle.com/code/henrychibueze/denoising-diffusion-probabilistic-model/notebook